International Journal of Computer Applications in Technology

ISSN online: 1741-5047 - ISSN print: 0952-8091
https://www.inderscience.com/ijcat

Pose estimation technology of electronic components based on point cloud segmentation algorithm

Wei Shen

DOI: 10.1504/IJCAT.2025.10074466

Article History:

Received:	25 February 2025
Last revised:	18 July 2025
Accepted:	11 September 2025
Published online:	17 February 2026

Pose estimation technology of electronic components based on point cloud segmentation algorithm

Wei Shen

School of Engineering,
Nanjing Normal University Zhongbei College,
Danyang, Jiangsu, China
Email: shenw1221@163.com

Abstract: In actual manufacturing environments, electronic components often face occlusion problems, which makes it difficult for traditional point cloud segmentation methods to estimate the pose of objects accurately. To address this challenge, this paper introduces the multi-scale feature learning capability provided by PointNet++ to extract deep collective feature information in local areas of different scales and understand the overall morphology of components in a global context. According to experimental analysis, under the same occlusion level, PointNet++ outperforms the PointNet model, the RANSAC (Random Sample Consensus) algorithm, and the voxelisation method Point-Voxel CNN in terms of segmentation accuracy. The pose estimation method of electronic components studied in this paper is highly applicable in actual mechanical manufacturing environments, can process large-scale data, and meets real-time requirements. It provides the theoretical basis and technical support for solving the positioning and assembly problems of components in actual industrial production.

Keywords: point cloud segmentation; pose estimation; PointNet++ Model; occlusion problems; mechanical manufacturing; random sample consensus.

Reference to this paper should be made as follows: Shen, W. (2026) 'Pose estimation technology of electronic components based on point cloud segmentation algorithm', *Int. J. Computer Applications in Technology*, Vol. 78, No. 2, pp.124–134.

Biographical notes: Wei Shen focuses his research on robot technology and machine vision. He has published four Chinese-language papers in Peking University's core academic journals, authored one authoritative textbook, and holds two invention patents, four utility model patents, as well as two software copyrights.

1 Introduction

With the rapid development of industrial automation (Pandey et al., 2024; Ajiga et al., 2024), the precise positioning and assembly of electronic components have become an important part of intelligent manufacturing (Zhang et al., 2024; Aldrini et al., 2024) and robot operation. This is mainly attributed to the fact that precise positioning and assembly can significantly improve production efficiency and product quality, reduce human error and in intelligent manufacturing systems, highly automated production lines rely on precise component positioning to achieve complex assembly tasks, while robot operations require precise control to complete fine work. As one of the core technologies, pose estimation technology aims to ensure that the automation system can operate efficiently and accurately by accurately calculating the spatial position and orientation of components. However, in practical applications, point cloud segmentation (Rosu et al., 2022; Xu et al., 2023; Xue et al., 2024) algorithms face many challenges, especially when dealing with electronic components with complex shapes or occlusion. For example, complex shapes lead to difficulties in feature extraction, missing parts of the information lead to the influence of segmentation accuracy, and traditional algorithms are insufficient in accuracy, robustness and computational efficiency. Traditional algorithms' shortcomings in accuracy, robustness and computational efficiency often lead to unsatisfactory pose estimation (Zheng et al., 2023; Dubey and Dixit, 2023; Fang et al., 2022). Therefore, improving the accuracy of point cloud segmentation and optimising computational efficiency has become an urgent problem to be solved in current research.

Aiming at the problem of inaccurate pose estimation of electronic components due to occlusion in actual application environments, this paper studies PointNet++. When electronic components are faced with occlusion problems, the local features around the occluded part are extracted. After extraction, the features of different areas are fused for

contextual analysis, and then the features of the occluded part are completed through feature aggregation. Finally, the robustness of the completed occlusion is enhanced through feature propagation. The model was trained and experimentally analysed, and the pose estimation method studied in this paper showed the highest segmentation accuracy (IoU=0.85, Dice=0.91) compared with the PointNet model, the voxelisation method Point-Voxel CNN and the RANSAC algorithm. Different degrees of occlusion were designed, and PointNet++ could maintain a high pose estimation accuracy under different occlusion ratios. In addition, when compared with the above three models or algorithms under high occlusion conditions, PointNet++ showed the best pose estimation accuracy. The experiments designed to analyse the real-time performance and computational efficiency of PointNet++ in simulated actual application environments show that PointNet++ can achieve a good balance between reasoning efficiency and resource usage, and performs well in the case of large-scale data. The significance of this study is to design a method that can significantly improve the accuracy of pose estimation of electronic components in complex manufacturing environments when facing occlusion problems. It provides strong support for the precise positioning and assembly of electronic components in actual industrial production, and promotes the application and development of intelligent manufacturing and robotics technology to a certain extent.

The main contribution of this paper is to accurately segment point cloud data, effectively extract electronic components from complex scenes and provide a high-quality data basis for subsequent posture estimation. The technology combines point cloud segmentation and posture estimation algorithms to accurately estimate the three-dimensional position and posture of electronic components, and improves the accuracy and efficiency of robot grasping, assembly and other operations.

2 Related work

At present, many researchers have proposed a variety of point cloud segmentation algorithms to solve the problem of pose estimation of electronic components. To solve the problems of traditional pose estimation methods, such as the need for a large number of markers and unreliable output, Biderman et al. introduced the 'Lightning Pose' efficient pose estimation method, which improves the label penalty, network architecture and pose prediction (Biderman et al., 2024). Bauza et al. (2023) introduced an object-specific Tac2Pose model for estimating tactile poses from the first touch of a known object; that is, given the geometry of the object, the model can estimate the probability distribution of possible object poses given a tactile observation. Hoang et al. (2022) used an end-to-end learning method to estimate the pose of electronic components given the raw point cloud input, and learned more advanced features by exploiting the dependencies between object parts and object instances. In Elsisi et al. (2023), a new robust Kalman Filter (KF) is introduced for AGV state estimation using the Huber loss function. A statistical method called M estimation is used to robustly solve the regression problem and establish the equivalence between KF and a specific least squares regression problem. The method solves the limitations of insufficient real-time performance and robustness encountered in estimating the attitude of non-cooperative spacecraft during in-orbit missions, Yuan et al. (2024) proposed a new feature point distribution selection learning method. This method uses a non-coplanar critical point selection network with uncertain prediction, which is groundbreaking in the ability to accurately estimate the attitude of non-cooperative spacecraft. However, existing point cloud segmentation algorithms still face many challenges when facing complex practical application scenarios.

To address the above issues, Park and D'Amico et al. (2024) proposed a spacecraft pose network based on convolutional neural networks, which was used to estimate the pose of non-cooperative spacecraft across domain gaps. In addition, the study also introduced online domain refinement, which can refine the parameters of the target domain image in the normalisation layer when the network is deployed. Ying et al. (2022) proposed a method for detecting connections and estimating 6DoF pose. The specific approach is to use a high-precision 3D digital instrument to capture the point cloud of electronic components, and combine several methods, such as deep learning and registration, for data processing. Liu et al. (2023) proposed a robot continuous grasping system. The system uses accurate category-level 6-dimensional grasping pose information to achieve end-to-end robot grasping of objects in three-dimensional space. The core of the system is to achieve pose estimation in space. Given the problems of motion blur of the image caused by the influence of waves on the water surface of the unmanned ship during the landing stage of the UAV, and the low accuracy and poor robustness of obtaining the relative position and posture of the UAV, Ge et al. (2024) proposed a 6D target posture estimation algorithm based on multi-model key point weighted fusion to improve the accuracy and robustness of posture estimation. The robot's operating environment is complex, and the distribution of materials is random, which leads to low recognition and positioning accuracy of the robot's target posture and poor real-time performance. For this reason, Li (2024) proposed a robot target posture recognition method based on the improved particle swarm algorithm-BP neural network (PSO-BP).

However, although these methods have made progress in some scenarios, most methods still fail to maintain high accuracy while taking into account computational

efficiency. Therefore, existing methods still face the trade-off between real-time performance and accuracy in practical applications.

3 Data pre-processing and point cloud input

Pre-processing point cloud data can ensure the accuracy of point cloud segmentation and pose estimation. The noise of point cloud data comes from many factors. It mainly includes the error of the sensor itself, environmental interference, the difference in surface reflection characteristics of objects and mechanical vibration in the process of data acquisition. In order to reduce the impact of noise on subsequent segmentation, noise removal is first performed. The method used is to remove statistical outliers to achieve the denoising effect.

$$D_z = \frac{1}{k}\sum_{j=1}^{k}\left\| p_i - p_j \right\| \tag{1}$$

The distance between a point in the point cloud and its neighbouring points is calculated using formula (1). If the distance exceeds a preset threshold, the point is considered an outlier and is removed.

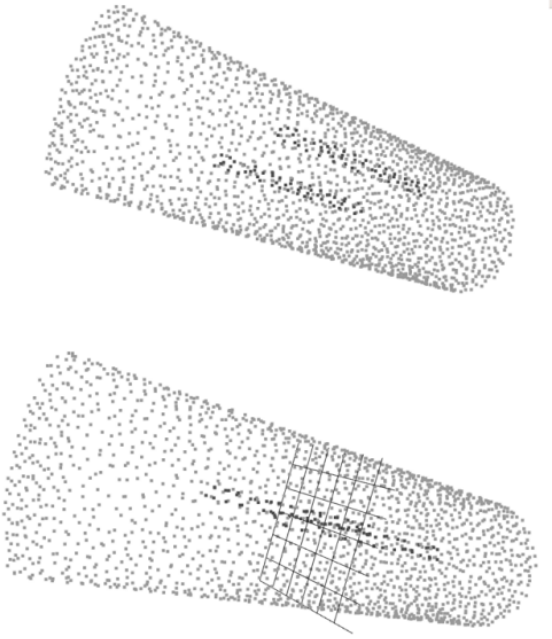
Point cloud data is usually large. Point cloud data is usually very large, mainly because high-precision scanning devices generate a large number of dense points to accurately describe the surface details of an object when collecting data. To ensure the integrity of the shape and structure of an object, a sufficient number of points need to be collected, which further increases the amount of data. Downsampling the input point cloud data can improve computational efficiency and reduce redundant data. The specific method used is voxel grid downsampling, which sets the size of a voxel grid, averages or represents the point cloud within each voxel and maps each point to its voxel centre.

$$P_c = \frac{1}{n}\sum_{l=1}^{n} p_l, \forall p_l \in V_c \tag{2}$$

V_c represents the voxel where the point p_l is located.

The sampling voxel grid downsampling performs the downsampling operation on the point cloud. Volume grid is a method of three-dimensional space division. It divides the continuous three-dimensional space into countless small cubes (i.e., volume elements). Each volume element represents a fixed area in the space, and the points in the point cloud data are assigned to the corresponding volume elements according to their coordinates, so as to realise the reduction sampling of the point cloud. Grid division is the main operation. Figure 1 is a schematic diagram of the grid division operation of a cylindrical resistor in three-dimensional space, and one of the cross-sections is selected for division.

Figure 1 Schematic diagram of grid division



The scale, position and direction of point cloud data are often inconsistent, so the input point cloud data is normalised.

$$P_k' = \frac{p_k - p_{centroid}}{r_{\max}} \tag{3}$$

$p_{centroid}$ represents the centroid of the point cloud data, and $r_{\max}$ represents the maximum radius of the point cloud.

The point cloud data is translated and rotated so that it is also in a standard coordinate system. The point cloud processed by the rotation matrix and translation vector is:

$$p_o' = Rp_o + t \tag{4}$$

The normal information of the point cloud is of great significance for point cloud segmentation and pose estimation. Because the normal can reflect the directionality and geometric features of the surface of an object, it helps to distinguish different surfaces and boundaries. In segmentation, the normal can assist in identifying the contour of an object; in pose estimation, the normal information can help determine the spatial position and direction of an object more accurately. The normal is estimated by calculating the covariance (Ledoit and Wolf, 2022).

$$C = \frac{1}{e}\sum_{j=1}^{e}\left(p_j - p_i\right)\left(p_j - p_i\right)^T \tag{5}$$

The point cloud input format of PointNet++ is:

$$p_v = \left[x_v, y_v, z_v, n_x, n_y, n_z\right] \tag{6}$$

Among them, x_v, y_v, z_v the first represents the spatial coordinates of the point, and n_x, n_y, n_z the second represents the normal vectors of the point.

4 Design of PointNet++

In the process of processing point cloud data, electronic components often face the problem of occlusion, and traditional point cloud segmentation methods often cannot effectively extract the features of electronic components when occlusion occurs. To address this problem, this paper introduces PointNet++ (Sorokin et al., 2024; Akbulut and Fevzi, 2025; Hu et al., 2024). According to the multi-scale feature learning mechanism in the model, the segmentation accuracy is improved by learning features of local areas of different scales and combining the features of the visible areas around the electronic components, thereby optimising the occlusion problem.

PointNet++ introduces the concept of hierarchical network structure, which means processing point cloud data of different scales through local feature extraction layers. The main advantage of multi-scale feature learning is that it can extract unique spatial information for each small local area in the point cloud, and then process it layer by layer to gradually integrate the surrounding larger range of information, thereby maintaining strong local features. PointNet++ is a deep learning model specifically designed for processing 3D point cloud data. It employs a hierarchical network structure, utilising local feature extraction layers to process point cloud data at various scales, enabling multi-scale feature learning. This approach effectively extracts spatial information from each small local region of the point cloud and integrates these features layer by layer, thereby enhancing the accuracy of segmentation and pose estimation.

The core steps of local area feature processing are as follows:

1) *Constructing the local neighbourhood of the point cloud*: For each point in the given point cloud data, the *K*-nearest neighbour method is used to extract its local neighbourhood information:

$$N(p_i) = \left\{ p_j \mid d\left(p_i, p_j\right) < \varepsilon \right\} \tag{7}$$

$d\left(p_i, p_j\right)$ represents the distance between points p_i, p_j.

2) *Extracting local features*: For each neighbourhood constructed in step (1), PointNet++ obtains the features of each area through a Multilayer Perceptron (MLP) (Yaylacı et al., 2023; Zhang et al., 2023; Ghimire et al., 2023). By transforming the features of the domain points through the neural network layer of the MLP, the features of each point in its local area can be calculated.

   The calculation formula is:

$$f_{local}\left(P_i\right) = MLP\left(P_i\right) \tag{8}$$

3) *Aggregate features and fuse context*: The analysis of features is not limited to a single local area, but also combines more areas for context inference. When analysing features, PointNet++ not only considers the information from the current local area but also incorporates information from a broader surrounding area. By integrating features from multiple regions, the model can perform contextual reasoning, thereby gaining a more comprehensive understanding of the global structure of point cloud data, thus enhancing the accuracy and robustness of feature analysis. A hierarchical pooling strategy is designed to aggregate information of different scales on PointNet++. The purpose of aggregation is to achieve multi-level feature fusion based on local feature learning.

$$f_{global} = MaxPool\left(f_{local1}, f_{local2}, ..., f_{localN}\right) \tag{9}$$

This operation aggregates local features of different scales into global features f_{global}.

When a part of an electronic component is occluded, PointNet++ can supplement the missing information by aggregating smaller local areas. In the process of local feature extraction, part of the point cloud data may be lost due to occlusion, but the model can recover the features of the complete object from the unoccluded part through more detailed local feature extraction. In addition, PointNet++ uses an efficient feature propagation mechanism, so that the local features of the occluded part can be propagated through the neighbourhood points, further enhancing the robustness under occlusion. PointNet++ In addition to its efficient feature propagation mechanism, it also possesses multi-scale feature learning capabilities, enabling the capture of local features at different scales; the hierarchical network structure allows for information fusion layer by layer, enhancing global understanding; it performs excellently when processing large-scale data, combining high precision with real-time performance, meeting the requirements of industrial applications.

Figure 2 compares the segmentation of a circular electronic component commonly used in industry using different segmentation methods. The upper part of the figure is the original point cloud of the circular ring. The grey points are the points in the electronic component, and the red points are the occluded parts. The selected segmentation methods include PointNet++ studied in this paper and a segmentation method based on model fitting, RANSAC (Wu, 2023; Kuçak, 2022). From the segmentation results in the figure, PointNet++ can use the local feature extraction of electronic component point cloud data to restore the complete device point cloud features from the unoccluded part. In comparison, the RANSAC method still has occlusions after segmenting the occluded devices, and cannot solve this problem well.

After the PointNet++ network segments and extracts features of electronic components, the pose of the electronic components is estimated through an additional fully connected layer. The network outputs the pose information of the electronic components, which includes the position and rotation angle of the electronic components.

Figure 3 is the PointNet++ network architecture designed in this paper.

Figure 2 Comparison of segmentation methods (see online version for colours)

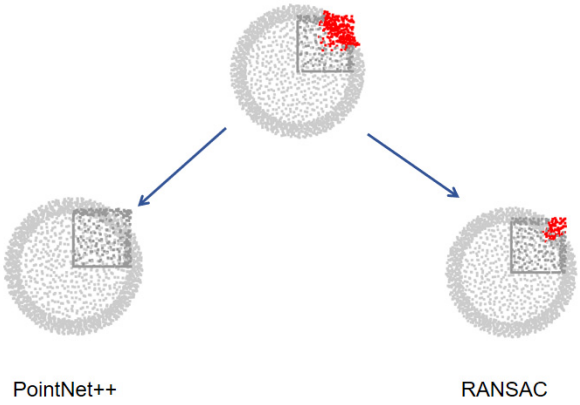
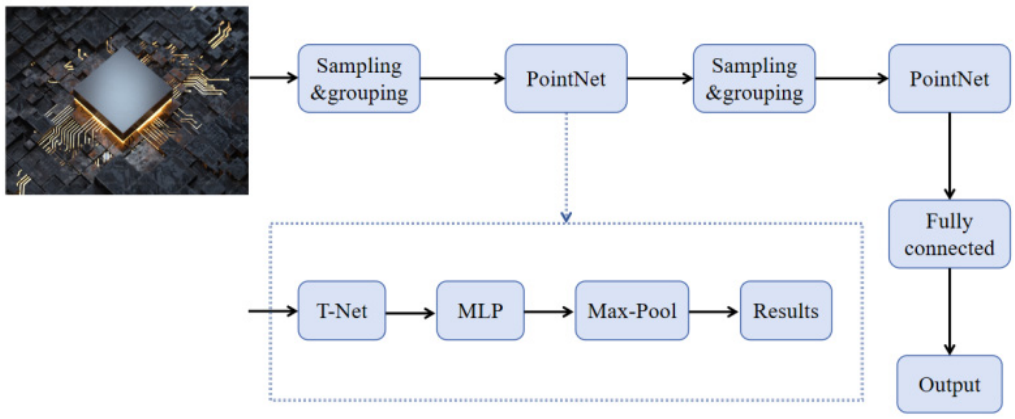


Figure 3 PointNet++ network architecture (see online version for colours)



In the network architecture, electronic components are processed through multiple point cloud data sampling layers and PointNet (Ding et al., 2023) layers, and finally, the pose estimation results are output through a fully connected layer.

5 Pose estimation

The main goal of pose estimation is to predict the 3D spatial position and orientation of an object relative to a camera or work platform. Pose estimation can be considered as a regression problem, which predicts the position vector and rotation matrix of an object based on the input point cloud data. Pose estimation can be regarded as a regression problem because its core lies in learning the continuous mapping relationship from point cloud data to the three-dimensional spatial position vector and rotation matrix of an object through the model. This mapping is continuous and numerical, aiming to predict specific parameter values rather than discrete categories, thus conforming to the characteristics of a regression problem.

The position vector represents the position of an electronic component in 3D space.

$$t = \left[t_x, t_y, t_z \right] \tag{10}$$

t_x, t_y, t_z represent the translation of the electronic component along the x-, y- and z-axes in three-dimensional space.

After feature extraction by the PointNet++ network, the prediction process of the position vector of the electronic component is:

$$t = W_t \cdot f + b_t \tag{11}$$

Among them, W_t the weight matrix of the translation vector represents the weight matrix of the translation vector, and b_t represents the bias term in the process of predicting the position vector.

Quaternions are used to represent the direction of rotation of electronic components in three-dimensional space, and quaternions are defined as:

$$q = \left[q_0, q_1, q_2, q_3 \right] \tag{12}$$

Among them, q_0 the first represents the scalar part of the rotation, and the other three represent the vector part.

Quaternion has advantages in avoiding gimbal contraction, computational efficiency and interpolation smoothness. It is the most commonly used method in rotation representation. In pose estimation, in addition to using quaternion representation, Euler angles can also be used for representation. Euler angle is used to describe the three angular parameters of a rigid body's rotating attitude in three-dimensional space. Specifically, it refers to the three angles of rotation around a fixed axis, namely pitch angle, yaw angle and roll angle. Table 1 compares quaternions and Euler angles.

Table 1 Comparison of quaternions and Euler angles

Characteristic	*Euler angles*	*Quaternions*
Representation	Three angles (pitch, yaw, roll)	4 components (1 scalar + 3 vectors)
Computational efficiency	Low	High
Avoiding Gimbal Lock	No	Yes
Interpolation smoothness	Poor (may cause discontinuity)	Good
Physical meaning	Rotation angle	Rotation angle and rotation axis

In the pose estimation task, using quaternions to represent rotation can effectively improve the stability and robustness of the model and avoid the shortcomings of Euler angles.

In the pose estimation of electronic components, the global feature vector is extracted by PointNet++, and the quaternion prediction process of rotation is expressed as:

$$q = Sigmoid\left(W_q \cdot f + b_q\right) \tag{13}$$

Among them, W_q represents the weight matrix of the rotation quaternion, b_q represents the bias term in the prediction process of the rotation quaternion and the Sigmoid activation function is used to ensure that the output value of the quaternion falls within the valid range.

The process of obtaining the rotation matrix R from the quaternion q is as follows:

$$R = \begin{bmatrix} 1-2\left(q_2^2+q_3^2\right) & 2\left(q_1q_2-q_0q_3\right) & 2\left(q_1q_3+q_0q_2\right) \\ 2\left(q_1q_2+q_0q_3\right) & 1-2\left(q_1^2+q_3^2\right) & 2\left(q_2q_3-q_0q_1\right) \\ 2\left(q_1q_3-q_0q_2\right) & 2\left(q_2q_3+q_0q_1\right) & 1-2\left(q_1^2+q_2^2\right) \end{bmatrix} \tag{14}$$

The rotation matrix R is used to describe the orientation of electronic components in three-dimensional space.

The objective function of pose estimation is set as:

$$L_{pose} = \lambda_1 \cdot L_{translation} + \lambda_2 \cdot L_{rotation} \tag{15}$$

$L_{translation}$ Represents the position error of the electronic component, $L_{rotation}$ represents the rotation error and λ_1, λ_2 represents the weight coefficients for controlling the position and rotation errors, respectively.

The position error is defined as:

$$L_{translation} = \left\|\hat{t} - t\right\|_2 \tag{16}$$

The position error represents the L2 norm difference between the predicted displacement vector and the true displacement vector.

The rotation error is measured by calculating the angular error between the rotation matrix and the true rotation matrix.

$$L_{rotation} = \arccos\left(\frac{Tr\left(R^T\hat{R}\right)-1}{2}\right) \tag{17}$$

In order to further improve the pose estimation progress when the occlusion problem occurs, this paper combines multi-view point clouds and reduces the impact of occlusion by inputting point cloud data from multiple perspectives. By inputting multi-viewpoint point cloud data, the model can capture object information from different perspectives. When there is occlusion in one perspective, the complete data of other perspectives can provide a supplement. After integrating multi-viewpoint features, the object features can be extracted more comprehensively and accurately, reducing the interference of occluded parts on attitude estimation, so as to reduce the occlusion effect. The multi-view integration method is as follows:

$$P_{multi} = \sum_{i=1}^{N} P_i \tag{18}$$

The implementation of this strategy enables the PointNet++ network to obtain more complete point cloud information from different perspectives and reduce the pose estimation error caused by occlusion.

6 Experiment design

The experiments in this paper are conducted on the MVTec ITODD public data set, which is mainly aimed at object detection and pose estimation, focusing on electronic components such as industrial parts. This data set contains occlusion conditions of various electronic components, which is very helpful for testing the effect of PointNet++ point cloud segmentation under occlusion in this paper's experiments and verifying the pose estimation effect. Given the specific application scenarios in the experiment, this paper has performed relevant data enhancement operations on some devices in the data set to better meet the research needs of this paper. In the experiment, the data set is divided into a training set, test set and validation set, with the proportions of the three being 60%, 20% and 20%, respectively.

The experiment in this paper is carried out in the environment in Table 2.

Table 2 Experiment environment configuration

Configuration items	*More information*
CPU	Intel i7-9700K
GPU	NVIDIA RTX 2080 Ti
Memory	32GB
Hard disk	1TB SSD
Operating system	Ubuntu 18.04
Deep learning frameworks	PyTorch 1.9
Python version	Python 3.8
CUDA version	CUDA 11.0
Video memory	11GB
Training batch size	16
Learning rate	0.001
Optimiser	Adam

This experiment uses an Intel i7-9700K eight-core processor, paired with an NVIDIA RTX 2080Ti graphics card, 32 GB of memory, and a 1TB SSD hard drive to ensure efficient processing and fast access to large-scale data. The operating system uses the stable Ubuntu 18.04 version, paired with the PyTorch 1.9 deep learning framework, which provides a good development environment and GPU acceleration support for PointNet++. CUDA 11.0 further optimises performance, and the graphics card's 11GB video memory ensures the ability to process large-scale point cloud data. The training batch size during model training is 16, the learning rate is set to 0.001 and the Adam optimiser is used for optimisation to ensure stability and efficiency during model training.

The experiment steps are as follows:

1) Before the experiment begins, the point cloud data from the data set is first loaded, and standardised preprocessing is performed, including denoising, rotation, translation and other data enhancement operations.
2) For the incomplete occlusion that may exist in the data set, artificial occlusion is simulated and the simulated occlusion data is also added to the training set and test set, so that the model can fully learn the incomplete occlusion point cloud information.
3) The processed point cloud data can be used to train the PointNet++ model, and the cross-validation method is used during the training process.
4) After training, the model is tested using data under occlusion, focusing on evaluating the model's pose estimation capabilities under different degrees of occlusion. By comparing with the actual pose data, the angle error and position error are calculated, with special attention paid to the performance changes of the model under occlusion.

7 Results presentation and analysis

7.1 Segmentation accuracy analysis

In the pose estimation of electronic components, segmentation is the first step, and the segmentation accuracy directly affects the accuracy of pose estimation. High-precision segmentation can ensure the accuracy of the device boundary, thereby improving the accuracy of pose estimation. By comparing the segmentation accuracy of different algorithms, we can evaluate the adaptability and robustness of the algorithms in different scenarios. In the experiment, the PointNet++ model studied in this paper is compared with the PointNet model, the RANSAC algorithm, and the voxelisation method Point-Voxel CNN (Sun and Lin, 2024). In the experiment, the segmentation accuracy analysis of different algorithms controlled the occlusion degree of electronic components at the same level. The segmentation accuracy was quantified using two indicators, IoU and Dice coefficient and the image analysis of the segmentation accuracy results was drawn.

From Figure 4, PointNet++ in this paper performs best in terms of IoU (0.85) and Dice (0.91), showing the high efficiency of the model in point cloud segmentation tasks, ensuring accurate segmentation of electronic components. In contrast, although PointNet also performs well in two indicators (only below PointNet), this result may be due to PointNet's lack of deep learning optimisation of local details, resulting in insufficient segmentation performance in the presence of occlusion. RANSAC's segmentation accuracy is the lowest among the four algorithms (IoU=0.65, Dice=0.75), which shows that the algorithm performs poorly in the face of occlusion, probably because the algorithm mainly relies on assumptions and random sampling and does not have learning capabilities. The IoU score of Point-Voxel CNN is 0.72, and the Dice score is 0.79. Its performance in segmentation accuracy is only slightly better than RANSAC. This is because Point-Voxel CNN is affected by voxelisation. Although it has certain advantages in processing point cloud data, it is easy to lose some details in the voxelisation process. In summary, PointNet++ benefits from the advantages of the deep learning framework and can learn the details and optimisation features of electronic components from point cloud data, and has higher accuracy when processing point cloud segmentation tasks.

7.2 Pose estimation under occlusion

Occlusion is a common challenge in actual industrial applications and operations. In order to evaluate the accuracy and robustness of the pose estimation of the PointNet++ model studied in this paper in real application environments, the pose estimation accuracy of electronic components under different occlusion conditions is analysed in the experiment. In order to simulate different occlusion situations, 10%, 20%, 30% and 40% occlusions were designed for the same electronic component relative to the entire device. The analysis indicators include the angle error and position error in pose estimation. Table 3 shows the results.

Figure 4 Comparison of IoU and Dice indicators (see online version for colours)

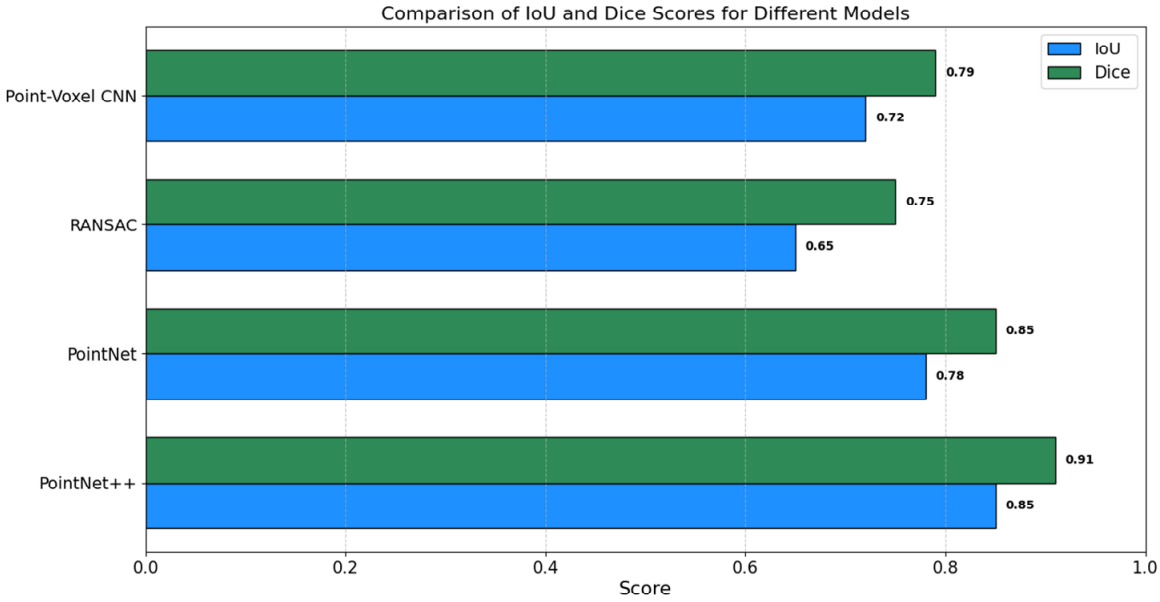


Table 3 Analysis of different proportions of occlusion

Occlusion ratio	*Position error (mm)*	*Angular error (°)*
0%	2.5	0.8
10%	3.1	1.1
20%	3.8	1.5
30%	4.6	2.2
40%	5.3	2.8

It can be seen from Table 3 that as the occlusion ratio continues to increase, the position error and angle error of PointNet++ in pose estimation show a trend of gradual increase. PointNet++ can still maintain a relatively stable pose estimation accuracy under different occlusion ratios: when there is no occlusion, the position error is 2.5 mm and the angle error is 0.8°. When the occlusion is 40%, the position error increases to 5.3 mm and the angle error increases to 2.8°, showing a smaller increase. Although the increase in occlusion ratio can lead to an increase in the pose estimation error of PointNet++ and affect the accuracy of pose estimation, the increase in error of pose estimation is small as the occlusion ratio continues to increase. PointNet++ can still maintain a low error in the case of high occlusion (40%), which fully proves that it can show a strong application effect in complex application environments.

In order to further analyse the effect of this study on pose estimation and more comprehensively evaluate the robustness and accuracy of PointNet++ in complex environments, the PointNet++ model is compared with the voxelisation method Point-Voxel CNN, the RANSAC algorithm and the PointNet model. Based on the above experiments, improvements are made and only high occlusion (40%) conditions are retained. This is because high occlusion conditions can more clearly demonstrate the reliability of the model or algorithm in practical applications. In addition to position and angle errors, the comparison indicators also include the accuracy and recall of pose estimation.

As can be seen from Table 4, in the case of high occlusion, PointNet++ shows the best pose estimation accuracy, with position error (5.3 mm) and angle error (2.8°), which are better than PointNet (6.2 mm and 3.5°), Point-Voxel CNN (8.7 mm and 5.0°) and RANSAC (10.3 mm and 6.1°). The emergence of this result is mainly attributed to the advantage of PointNet++ in capturing local features. Compared with PointNet, PointNet++ introduces more detailed learning of local areas through a hierarchical structure, enabling the model to estimate the pose of electronic components well, even in the case of high occlusion. At the same time, PointNet++ also showed the best performance in pose estimation precision (0.91) and recall rate (0.90), which shows that the model can accurately estimate the pose of electronic components even in the case of high occlusion and can also identify electronic components well. This result well demonstrates the advantage of PointNet++ in accurately estimating the pose of electronic components by combining local features.

Table 4 Comparison of different algorithms or models

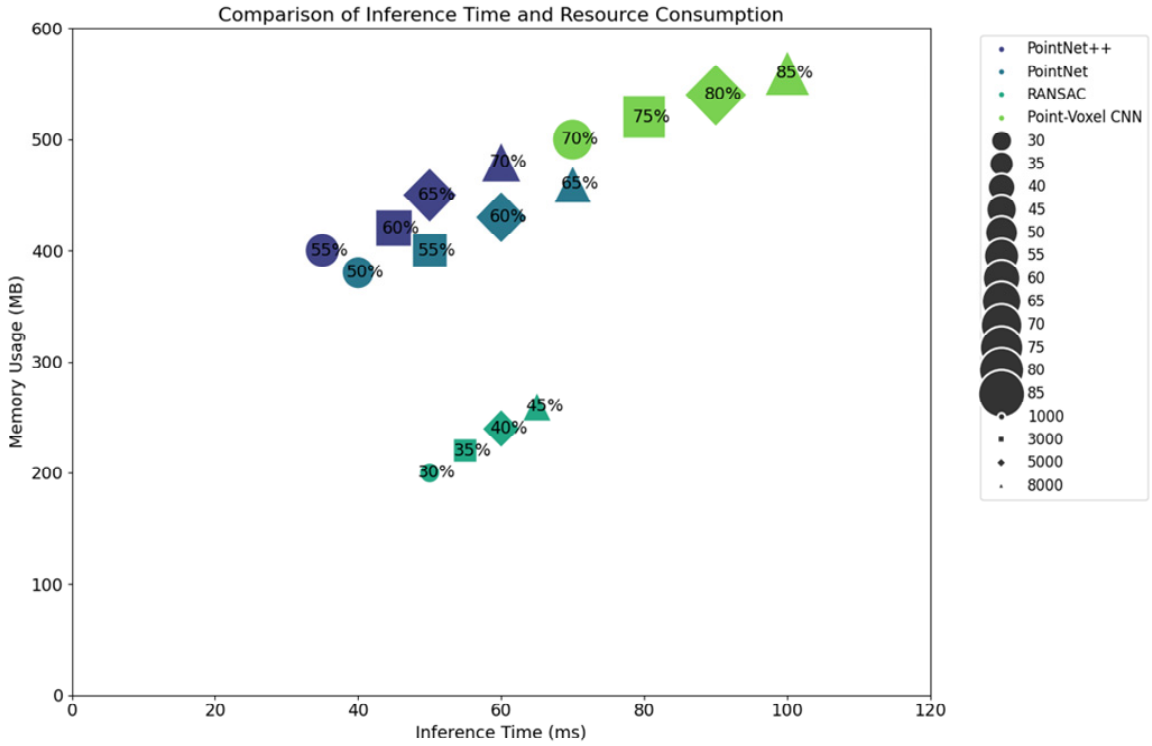
Algorithms/ Models	*Position error (mm)*	*Angular error (°)*	*Precision*	*Recall*
PointNet	6.2	3.5	0.85	0.83
PointNet++	5.3	2.8	0.91	0.90
RANSAC	10.3	6.1	0.75	0.72
Point-Voxel CNN	8.7	5.0	0.80	0.77

7.3 Real-time and computational efficiency analysis

In actual industrial applications, the pose estimation of electronic components must not only ensure high accuracy but also fast processing. Especially in real-time production lines, the calculation speed and resource consumption of the model directly affect the application's effect. In the experiment, the PointNet++ model is compared with the RANSAC algorithm, the PointNet model and the voxelisation method Point-Voxel CNN in terms of inference time and resource consumption. Different data amounts are designed for experiments (1000, 3000, 5000, 8000), and the experimental results are plotted as a comparative bubble chart.

Figure 5 shows a bubble chart comparing the inference time and resource consumption of four models or algorithms in the experiment. The *X*-axis represents the inference time, and the *Y*-axis represents the memory usage. The size of the bubble in the figure shows the GPU usage rate. The larger the bubble, the more GPU is used. Different models or algorithms are distinguished by different colours. From the bubble chart and experimental data analysis, PointNet++ has a relatively balanced performance in terms of inference time and GPU occupancy. The inference time increases with the increase in data volume, and the GPU occupancy increases steadily, which shows that it has good efficiency and resource utilisation when processing large-scale data. In comparison, RANSAC shows a lower growth trend in inference time and GPU usage, with minimal memory consumption and is more suitable for environments with limited computing resources. Point-Voxel CNN has higher inference time and memory usage, especially in the case of large amounts of data, where the inference time increases significantly. This may be related to the voxelisation method's distributed complexity and memory requirements. Compared with PointNet++, PointNet has higher inference time and GPU usage, but lower memory consumption and GPU resource requirements. Overall, PointNet++ can achieve a good balance between inference efficiency and resource usage, and performs well in the case of large-scale data.

Figure 5 Bubble chart of inference time and resource consumption (see online version for colours)



8 Conclusions

Given the difficulty of estimating the pose of electronic components in the face of occlusion in the actual industrial environment, this paper studies PointNet++. By using the local feature extraction function of PointNet++ and combining the feature aggregation analysis of neighbourhood points, the pose estimation accuracy of electronic components in the face of occlusion is improved. The main contributions of this paper are:

1) A solution based on PointNet++ is designed to solve the occlusion problem of electronic components in pose estimation. The multi-scale learning features are introduced, and the accuracy of pose estimation is significantly improved by combining feature extraction and feature fusion technology in local areas.

2) Through the feature aggregation and propagation of PointNet++, the robustness of the occluded area is enhanced, the feature information of the occluded part is effectively supplemented and the accuracy of pose estimation is further improved.

3) Experiment analysis verifies the superior performance of PointNet++ under different occlusion conditions, especially the segmentation accuracy and accuracy preservation under high occlusion conditions. In addition, it verifies that PointNet++ has a good balance between reasoning efficiency and resource usage, and can process large-scale data sets.

4) The research results provide a technical solution for the positioning and assembly of electronic components, which has practical value in the fields of intelligent manufacturing and industrial automation.

However, there are still some shortcomings in this study. First, the experiments in this paper are based on a specific data set, while the data in actual applications may contain more noise and diverse shapes. Second, for some extreme occlusions, such as more than 70% of electronic components being occluded, the model may not be able to fully recover the features of the occluded part. Finally, as a deep learning model, PointNet++ has weak interpretability of its internal mechanism. Future research can expand the training data set and enhance data processing, improve the pose estimation technology under computational occlusion and combine artificial intelligence technology to improve the transparency and interpretability of the PointNet model, further enhancing the credibility and controllability of the model in actual industrial applications.

Acknowledgement

This work was supported by National Natural Science Foundation of China (51875266).

Declarations

All authors declare that they have no conflicts of interest.

References

Ajiga, D., Folorunsho, S.O. and Ezeigweneme, C. (2024) 'The role of software automation in improving industrial operations and efficiency', *International Journal of Engineering Research Updates*, Vol. 7, No. 1, pp.22–35.

Akbulut, Z. and Karsli, F. (2025) 'DAPNet++: density adaptive PointNet++ for airborne laser scanning data', *Earth Science Informatics*, Vol. 18, No. 1, pp.1–24.

Aldrini, J., Chihi, I. and Sidhom, L. (2024) 'Fault diagnosis and self-healing for smart manufacturing: a review', *Journal of Intelligent Manufacturing*, Vol. 35, No. 6, pp.2441–2473.

Bauza, M., Bronars, A. and ARodriguez, A. (2023) 'Tac2pose: tactile object pose estimation from the first touch.' *The International Journal of Robotics Research*, Vol. 42, No. 13, pp.1185–1209.

Biderman, D., Whiteway, M.R., Hurwitz, C., Greenspan, N., Lee, R.S. and Vishnubhotla, A. et al. (2024) 'Lightning pose: improved animal pose estimation via semi-supervised learning, Bayesian ensembling and cloud-native open-source tools', *Nature Methods*, Vol. 21, No. 7, pp.1316–1328.

Ding, C., Zhang, L., Chen, H., Hong, H., Zhu, X. and Fioranelli, F. (2023) 'Sparsity-based human activity recognition with PointNet using a portable FMCW radar', *IEEE Internet of Things Journal*, Vol. 10, No. 11, pp.10024–10037.

Dubey, S. and Dixit, M. (2023) 'A comprehensive survey on human pose estimation approaches', *Multimedia Systems*, Vol. 29, No. 1, pp.167–195.

Elsisi, M., Altius, M., Su, S.F. and Su, C.L. (2023) 'Robust Kalman filter for position estimation of automated guided vehicles under cyberattacks', *IEEE Transactions on Instrumentation and Measurement*, Vol. 7.2, pp.1–12.

Fang, H-S., Li, J., Tang, H., Xu, C., Zhu, H. and Xiu, Y. (2022) 'Alphapose: whole-body regional multi-person pose estimation and tracking in real-time', *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 45, No. 6, pp.7157–7173.

Ge, Q., Li, K., and Zhang, X. (2024) 'A relative posture estimation algorithm for drones based on weighted fusion of multi-key point detection', J*ournal of Automation*, Vol. 50, No. 7, pp.1402–1416.

Ghimire, S., Nguyen-Huy, T., Prasad, R., Deo, R.C. and Salcedo-Sanz, S. et al. (2023) 'Hybrid convolutional neural network-multilayer perceptron model for solar radiation prediction', *Cognitive Computation*, Vol. 15, No. 2, pp.645–671.

Hoang, D-C., Stork, J.A. and Stoyanov, T. (2022) 'Voting and attention-based pose relation learning for object pose estimation from 3D point clouds', *IEEE Robotics and Automation Letters*, Vol. 7, No. 4, pp.8980–8987.

Hu, H., Tan, Q., Kang, R., Wu, Y., Liu, H. and Wang, B. (2024) 'Building extraction from oblique photogrammetry point clouds based on PointNet++ with attention mechanism', *The Photogrammetric Record*, Vol. 39, No. 185, pp.141–156.

Kuçak, R.A. (2022) 'The feature extraction from point clouds using geometric features and ransac algorithm', *Advanced LiDAR*, Vol. 2, No. 1, pp.15–20.

Ledoit, O. and Wolf, M. (2022) 'The power of (non-) linear shrinking: a review and guide to covariance matrix estimation', *Journal of Financial Econometrics*, Vol. 20, No. 1, pp.187–218.

Li, P. (2024) 'A robot target posture recognition method based on the improved PSO-BP algorithm', *Foreign Electronic Measurement Technology*, Vol. 42, No. 1, pp.7–12.

Liu, J., Sun, W., Liu, C., Zhang, X. and Qiang, F. (2023) 'Robotic continuous grasping system by shape transformer-guided multiobject category-level 6-D pose estimation', *IEEE Transactions on Industrial Informatics*, Vol. 19, No. 11, pp.11171–11181.

Pandey, N.K., Kumar, K., Saini, G. and Mishra, A.K. (2024) 'Security issues and challenges in cloud of things-based applications for industrial automation', *Annals of Operations Research*, Vol. 342, No. 1, pp.565–584.

Park, T.H. and D'Amico, S. (2024) 'Robust multi-task learning and online refinement for spacecraft pose estimation across domain gap', *Advances in Space Research*, Vol. 73, No. 11, pp.5726–5740.

Rosu, R.Al., Schütt, P., Quenzel, J. and Behnke, S. (2022) 'LatticeNet: fast spatio-temporal point cloud segmentation using permutohedral lattices', *Autonomous Robots*, Vol. 46, No. 1, pp.45–60.

Sorokin, M.I., Zhdanov, D.D. and Zhdanov, A.D. (2024) 'Conversion of point cloud data to 3D models using PointNet++ and transformer', *Programming and Computer Software*, Vol. 50, No. 3, pp.249–256.

Sun, Y. and Lin, Y. (2024) 'PVS-CNN: point-Voxel CNN optimized with substream sparse convolution', *Computer Applications and Software*, Vol. 41, No. 4, pp.135–141.

Wu, K. (2023) 'Creating panoramic images using ORB feature detection and RANSAC-based image alignment', *Advances in Computer and Communication*, Vol. 4, No. 4, pp.220–224.

Xu, Z., Liang, Y., Lu, H., Kong, W. and Wu, G. (2023) 'An approach for monitoring prefabricated building construction based on feature extraction and point cloud segmentation', *Engineering, Construction and Architectural Management*, Vol. 30, No. 10, pp.5302–5332.

Xue, W., Yang, Y., Li, L., Huang, Z., Wang, X. and Han, J. et al. (2024) 'Weakly supervised point cloud segmentation via deep morphological semantic information embedding', *CAAI Transactions on Intelligence Technology*, Vol. 9, No. 3, pp.695–708.

Yaylacı, M., Yaylaci, E.U., Özdemir, M.E., Öztürk, Ş. and Sesli, H. (2023) 'Vibration and buckling analyses of FGM beam with edge crack: finite element and multilayer perceptron methods', *Steel and Composite Structures, An International Journal*, Vol. 46, No. 4, pp.565–575.

Ying, C., Mo, Y., Matsuura, Y. and Yamazaki, K. (2022) 'Pose estimation of a small connector attached to the tip of a cable sticking out of a circuit board', *International Journal of Automation Technology*, Vol. 16, No. 2, pp.208–217.

Yuan, H., Chen, H., Wu, J. and Kang, G. (2024) 'Non-cooperative spacecraft pose estimation based on feature point distribution selection learning', *Aerospace*, Vol. 11, No. 7. Doi: 10.3390/aerospace11070526.

Zhang, J., Li, C., Yin, Y., Zhang, J. and Grzegorzek, M. (2023) 'Applications of artificial neural networks in microorganism image analysis: a comprehensive review from conventional multilayer perceptron to popular convolutional neural network and potential visual transformer', *Artificial Intelligence Review*, Vol. 56, No. 2, pp.1013–1070.

Zhang, X., Chen, H. and Wu, Y. (2024) 'Evaluation and validation of efficient hierarchical interconnection in multi-processor system on chip', *Cyber-Physical Systems*, Vol. 11, No. 2, pp.113–138.

Zheng, C., Wu, W., Chen, C., Yang, T., Zhu, S. and Shen, J. et al. (2023) 'Deep learning-based human pose estimation: a survey', *ACM Computing Surveys*, Vol. 56, No. 1, pp.1–37.